

VERIFYING A BALLOT USING PUBLIC KEY ENCRYPTION AND DIGITAL SIGNATURES

FIELD OF THE INVENTION

[0001] The invention relates to a method and system which provides security techniques to secure a voting system for use on a wide area network such as the global computer network, or on a local area network.

BACKGROUND OF THE INVENTION

[0002] The current approaches to provide a secure voting system on a wide area or local area network have traditionally emphasized purely cryptographic-protocol-based solutions to secure the voting. The work done to date is largely purely research into specific issues in secure electronic voting and does not address the practical application into a real-world system.

[0003] The prior art has approached worldwide area network, for example, on the global computer network known as the Internet, elections as an extension of secure, electronic commerce techniques without providing a comprehensive approach to the significant threats, vulnerabilities and risks that threaten the security, authenticity and reliability of such elections.

[0004] Internet elections must achieve the objectives of conventional elections, specifically: "democracy" (only registered citizens may vote once in any election), accuracy (votes may not be altered, forged or deleted), "privacy" (no one may know how anyone else voted or prove how they voted), and "verifiability" (everyone should be able to verify their own ballot, as well as the correctness of the entire election). Moreover, Internet elections address additional capabilities: "mobility" (individuals should be able to vote from any Internet-accessible location, at any legal time), "convenience" (voting should be simple and convenient for everyone), and "flexibility" (the technology should be applicable to all elections).

[0005] Internet voting systems must achieve these objectives in the face of both conventional election threats, and threats specific to automated, distributed computer systems, including: fraud, abuse of privilege (privileged individuals may act inappropriately), denial of service (computer services may be impaired or rendered inaccessible), software flaws, tampering (of recorded ballots or other data), malicious software, wiretapping (sniffing, modification or replay of communications), vote selling, or masquerading (by client/server computers or by people).

[0006] The prior art has taken several approaches towards Internet election systems. Electronic commerce or E-Commerce approaches rely on securing communications through encrypted connections and verifying individual identity only through weak or single-factor authentication (e.g., passwords). Encrypted Absentee Balloting systems emulate conventional absentee balloting by using ballots that are doubly encrypted using symmetric keys. Cryptographic protocols, particularly those relying on asymmetric or public-key cryptography hold the greatest promise but have been relegated largely to the academic community.

[0007] All these techniques have characteristic weaknesses. E-Commerce techniques secure only voter-system communications. They provide weak authentication and no (strong) mechanisms for achieving democracy, privacy, accuracy or verifiability. Encrypted Absentee Ballot systems provide better privacy, but provide no stronger mechanisms for democracy, accuracy or verifiability, and are vulnerable to abuse of privilege, potentially compromising the keys used to ensure privacy. Many of the cryptographic protocols developed to date are complex, making implementation verification difficult. Public-key cryptography and digital signature techniques promise strong authentication, accuracy and verifiability. However, the generation and distribution of public-private (secret) key-pairs, and the protection of private keys, makes these techniques difficult to apply to large-scale applications such as Internet voting.

[0008] Specifically, it has been recognized that existing private-key management techniques, i.e., PKI approaches, are not acceptable if they require the end users to buy, install or configure anything, or require any particular type of client device. Such considerations become particularly significant when there may be huge numbers of end users, for example, 100,000 to 1,000,000, such as may be the case in an election.

[0009] It is recognized that PKI technology supports the use of public-key cryptography by providing various means to generate public-private (secret) key pairs, protect access to the private (secret) key so that it can be used only by the individual with whose identity it is associated, and provide for distribution and convenient access to the public key. Common strategy involves generation of public/secret key pairs on a user's personal computer by an application such as a web browser, storage of the private key within a personal identification number (PIN) protected, encrypted key-store file on the user's personal computer (PC), and exportation of the public key to a certificate authority, for example, where it is wrapped in a digital certificate, such as an X.509 digital certificate, and made available for public access on a lightweight directory access protocol (LDAP) certificate directory service.

[0010] This approach has advantages in that it is convenient and the private keys never leave the user's PC. Disadvantages result, however, because even though the private keys are protected in PIN-based encrypted files, PCs offer little protection against key-store cryptanalysis by malicious software, and user mobility is impaired as it requires effort to export a private key so that it can be used on another platform.

[0011] An alternative strategy involves generation of public/secret key pairs on a secure platform rather than the user's PC, for example, a server. Yet still further, the approach involves storage of the private key, and perhaps other authentication-related data, on a storage device such as a password-encrypted floppy disk, or on a password-protected token/dongle, Java-ring (iButton) or smart card devices, as well as exportation of public keys and digital certificates as described with respect to the first approach.

[0012] One advantage of this approach is that the private key is more easily accessed from various computers, facilitating mobility, so long as there is a hardware interface for the private-key device. In addition, the private key is better protected within a removable device. Resultant disadvantages include the fact that the server must be verified not to make/leave copies of the private key. In addition, there must be some sort of hardware interface to allow retrieval of the key from the storage device, e.g., disk drive, USB port, iButton/smart card reader, etc. Yet still further, these hardware interfaces may be expensive, difficult to install/configure, and may not be universally available, thus inhibiting mobility.

[0013] The advantages and disadvantages of the above-identified two approaches have led to a third approach. In the third approach, generation of public/secret key pairs is done on a secure platform other than the user's PC, for example, a server. The secret keys are stored in encrypted form on the secure server and downloaded to the client over a secure network connection as needed to support authentication, digital signature or encryption operations. The server must authenticate the user by some non-PKI-based method before allowing the user to download their private key. In most cases, this will still require some authentication device.

[0014] While providing further refinements and including advantages such as convenience and mobility being improved because private keys are always available from a network server, and no hardware interfaces are required on the client device, significant disadvantages still remain.

[0015] Initially, it is noted that there is a need for a secondary strong authentication technique that requires hardware token support, e.g., SecureID. In addition, such hardware tokens incur additional expense, and private keys are stored on a secure server using one or more keys known to the server, thus requiring the server to protect access to and use of these keys.

[0016] In all three approaches described, retrieval of private keys from local/network storage is required for use within the memory of a PC or thin-client device, and exposes the key to potential compromise. Only tokens with processors can perform the necessary cryptographic operations without exposing the private key. None of the approaches offer sufficient security, mobility and convenience at a sufficiently low cost to make public/private key cryptographic services, e.g., authentication, non-repudiation, encryption, attractive for applications with a potentially large number of users such as in the case of an election with users numbering from anywhere between 100,000 to about 10,000,000.

[0017] In order to make public/private key cryptographic services feasible for such applications, there must be a low-cost way to generate public/secret keys, while providing secure storage for and access to those keys without requiring special hardware or inconvenient procedures. These advantages and other advantages are provided by the system and method described herein, and numerous disadvantages of existing techniques are avoided.

SUMMARY OF THE INVENTION

[0018] In accordance with one aspect of the invention, there is provided a method for assisting a user in verifying a cast ballot stored in a server. The method includes forming a digital signature of the cast ballot using a private key of the server, associating the cast ballot with such digital signature with a vote serial number (VSN), then forming a confirmation token comprising the digital signature of the cast ballot using a private key of the server and VSN. The token is then made available to a user. When the user wishes to individually verify that his vote has been recorded correctly, the user sends the token to the server where the VSN and signature of the cast ballot using a private key of the server are extracted from the token. For VSN equal to the extracted VSN, the received digital signature of the cast ballot using a private key of the server is compared to a stored digital signature of the cast ballot using a private key of the server. If the comparison shows equivalence between the data compared, determining that B_{cast} is verified.

[0019] As described herein, the terms "individual", "user", "client", and "voter" are used interchangeably, and refer to a person on their own terminal which can be a personal computer or other like device on which voting in accordance with the method and system herein is achieved.

[0020] In a more specific aspect, the confirmation token further comprises a digital signature of an aggregation comprising the associated cast ballot and VSN using the server's private key. The additional token element is extracted from the token returned to the system and the cast ballot is verified only upon the additional condition that received element is equivalent to its stored sibling.

[0021] Yet still further, the method involves recording in the server's data store the server's digital signature of the ballot to allow verification at the server that all of the ballots cast have not been tampered with.

[0022] In a more specific application, while the ballots are initially described as being generated as an HTML document, to provide additional security, in an alternative form the ballot can be generated in bit-map form such that the intentions of the voter or individual voting may be determined by monitoring the (x, y) coordinates of their mouse-clicks on the ballot bit-map. This provides enhanced security, since to read bit-map documents requires advanced optical

character recognition technology, which in turn requires and imposes a large computing power overhead, thus making malicious hacking into the system significantly more difficult and detectable.

[0023] Yet still further, the server is further configured for receiving the ballot choices and creating a data element from the electronic signature of the individual's ballot choices, the server's electronic signature of the individual's ballot choices, and the vote serial number to allow recording of the ballot choices in the data store and retained therein as a vote. This data element is then digitally signed using the server's private key to ensure its integrity and authenticity.

[0024] In a further aspect, the server is programmed for confirming retention of the vote at the data store by signing the individual's signature of the ballot choices, the server's signature of the ballot choices and the vote serial number, and thereafter transmitting the signed confirmation to the individual who submitted the ballot.

[0025] Yet still further, the system provides that the server is programmed for recording in the data store the server's digital signature of the ballot choices for allowing verification at the server that all of the ballots cast have not been tampered with.

[0026] In a still further aspect, the system is capable of generating the ballot as either a bit-map or an HTML document. The advantage of the bit-map document is that tampering or hacking becomes more difficult because bit-map documents require optical character recognition as previously discussed, such that malicious hacking is not easily achieved without detection.

BRIEF DESCRIPTION OF THE DRAWINGS

[0027] Having thus briefly described the invention, the same will become better understood from the following detailed discussion presented herein, with reference to the appended drawings, in which:

[0028] Fig. 1 is a system-level architectural view of how an operational system implementing electronic voting over a network, such as the global computer network known as the Internet, can be configured;

[0029] Figures 2 and 3 are diagrams illustrating the flow of the electronic ballot and associated data elements between an individual or client side, and a server side, managing the electronic voting method;

[0030] Fig. 4 is a block diagram showing in greater detail the components, and data information exchange illustrated in Figs. 1, 2, and 3; and

[0031] Fig. 5 is an architectural diagram showing in greater detail how a system and method as described herein could be deployed on a broad basis on a network such as the global computer network known as the Internet.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0032] Fig. 1 is a system-level architectural view of how an operational system and method as described herein might be implemented. The client personal computer 11 or PC 11, typically a personal computer running an operating system such as Microsoft Windows™, which may optionally include a smart card reader 13 connected thereto, although this is not required. The PC 11 includes network access capability which can be to the local area network, or to a global computer network 17. In the case of a global computer network 17, the PC 11 would have access to the global computer network 17 through, for example, an Internet service provider (ISP) 15, or alternatively, access to other parts of the network might be through a direct cable modem, or a local, or a network attachment to the global computer network 17.

[0033] On the server side, there is provided a primary interface to the entire system through a web server 19 in a conventional manner which is well known to those of ordinary skill in the art, and which is common to most electronic commerce architectures. The web server 19 interoperates with a certificate authority 21, which can be for example, an enterprise server suite such as that available from Netscape Corporation. The certificate authority 21 uses a certificate-generating system, such as one like that available commercially under the name iPlanet, which is conventional, and which is one of many alternatives which can be used to implement the functionality described herein.

[0034] In addition, the web server 19 can also interoperate and be connected to a back end application or database server 23 and an LDAP certificate directory server 25.

[0035] With respect to the LDAP certificate directory server 25, by way of explanation, it can provide a certificate such as an X.509 certificate, which is a text file that is used to contain information about an individual, together with an encoding of the individual's public encryption key to the system described herein. The certificate file is then digitally signed by the certificate authority 21 that issued it. If it is desired to use the public keys of a number of individuals in an open context, there has to be a way of accessing the X.509 certificates. This can be done by placing them on a lightweight directory access protocol server (LDAP) such as the LDAP certificate directory server 25 shown, through which access to those certificates can be provided. For example, LDAP servers are provided commercially through a number of entities, and it is possible to obtain from such servers another party's public key certificate, for example, if it is desired to provide digitally signed, encrypted electronic mail.

[0036] More specifically, LDAP is a protocol on a data architecture for storing name-value pairs and on an LDAP certificate directory server 25 there would be stored certificates, such as X.509 certificates bound to each individuals' name.

[0037] With respect to the application database server 23, it is also conventional and well known to those of ordinary skill in the art. Such an application database server 23 can be implemented using Java Servlets or Enterprise Java Beans (EJBs), which are typically small packets of functionality to make it easy and more efficient to use the more sophisticated Java implementation of web server 19 functionality. In operation, the web server 19 receives a request for a certain web page and in that web page is a reference to some call onto a piece of Java functionality that is written as a Java Servlet. An architecture known generally as the application database server 23 architecture allows the web server 19 to call functionality in the database server using Java Servlets or EJBs 23. The application/database server 23 is typically a separate machine that is specifically optimized to support the invocation of that kind of functionality, encoded, for example, as Java Servlets. . More particularly, the application database server 23 allows invoking of precompiled and packaged functionality that is written in the form of Java Servlets, Ems, or some other type of reusable component.

[0038] In the case of the present system and method, the reusable functionality will involve, for example, delivery of a ballot, which can be either in the form of an HTML or XML document, or for enhanced security as will be described hereafter, in the form of a bit-map.

[0039] Thus, when a user interacts with the web server 19 from their PC 11, a Servlet, or one or more EJBs, processes the request for the user to register with the system, including a request to cast a ballot. The Servlet or Ems that are invoked present the ballot to the user, and handle the interaction with the user in the course of casting the vote.

[0040] If it is desired to look at the results of an election, another Servlet or a different set of EJBs can be invoked to support that functionality. As may be appreciated by those of ordinary skill in the art, fine-grained packaging of functionality can be defined depending on the capabilities to be implemented as described hereafter.

[0041] In one implementation, a Registrar of Voters 27 may also be tied into the system and would be responsible for the registration of individuals who are legitimately, legally allowed to vote in a particular county. One way of conducting the registration is the conventional interacting in person with the Registrar of Voters, or through conventional mail. It is contemplated herein that such registration could be accomplished over the network, such as the global computer network 17, in a manner that it may be desired to have an electronic interface to the Registrar of Voters 27.

[0042] For example, one method of establishing the interface may be with a local Division of Motor Vehicles, particularly because of recent trends toward making the Division of Motor Vehicles responsible for identification and authentication of individual entities, and the Registrar of Voters appears to be an organization that might be able to use such capability.

[0043] As will also be appreciated, in the case of implementing an interface with the Registrar of Voters 27, some kind of X.509 certificate exchange as discussed previously could also be implemented, and the use of certificates would serve to authenticate any sort of interaction with the Registrar of Voters 27. In addition, if an individual submitted a registration or request, it could be digitally signed and thus would require the individual's public key certificate in order to validate the signature.

[0044] In all cases, it would be required to digitally sign some electronic document, and the digital signature would have associated with it a well-known identity which can be looked up on a server such as an LDAP certificate directory server 25 to obtain the public key certificate

corresponding to the claimed identity, and to then verify that the digital signature is actually a signature that could only have been generated by the individual having the claimed identity.

[0045] This is all standard public key infrastructure technology and well known to those of ordinary skill in the art. In the case of the Registrar of Voters 27, this could be done through a certified agent for registering the voters, and there might be several methods by which such agents could be implemented and authorized by the Registrar of Voters 27. Irrespective of what system is implemented, the agent would be acting as a front end to the Registrar of Voters 27, and proxying the registration to them, and so, it would have to be legally authorized to do so.

[0046] While on the server side a number of different individual functional components are shown, it will be appreciated as an alternative that all the functions can be implemented on a single server, or depending on the size of the system, the functions can be allocated to separate servers which themselves may be replicated. This is an alternative architectural approach well known to those of ordinary skill in the art, and only for the sake of clarity has the system been shown in Fig. 1 as separate boxes, which could be co-located on a single machine or duplicated or replicated on multiple boxes depending on the load and degree of redundancy and tolerance desired.

[0047] Figures 2 and 3 illustrate in block diagram form the data and information flow which will occur in a system and method as described herein. The components of data flow are shown divided by a dashed line which separates the client 41 side which relates to the individual and the individual's PC, and from the server 43 side.

[0048] A ballot on the client 41 side has been delivered from the server 43 side and can be some form of electronic document. The document can take various forms but for purposes of the description herein, the ballot is assumed to be an HTML form document, although more secure type and untamperable documents such as bit-map representations can also be deployed in accordance with the system and method.

[0049] One difficulty in building a system and method for such voting is being able to verify that only a single individual can cast a ballot, that they can cast only one ballot, and that it cannot be tampered with or undetectably altered or detected in any manner. Figures 2 and 3 illustrate

the relationships between data elements in an implementation for being able to achieve these goals.

[0050] The ballot is delivered, for example, to the home PC 11 through the web server 19 from the application database server 23, all of which were previously described with respect to Fig. 1. Using the home PC 11, through a web browser, an individual can go through and make selections on the ballot, and after pressing enter or casting the ballot, the PC 11 transmits back a subset of that form consisting of the cast ballot B_{cast} 51 and the voter's private digital signature of B_{cast} , i.e., $DS(B_{\text{cast}}, v)$.

[0051] Thus as is shown in Fig. 2, in the top arrow line from B_{cast} 51, the choices are delivered as a representation to the server side 43. The cast ballot B_{cast} 51 is structured in a way that it is easy to read the response values and determine which issues or offices were being voted on, and what the selections were. Those values are parsed and stored into a relational database 55, the structure of which will be described hereafter with reference to a separate figure. When the ballot is received on the server side 43 the server computes $DS(B_{\text{cast}}, s)$ 53, a digital signature of the cast ballot using the server's 43 private key.

[0052] Throughout this disclosure in labels such as $DS(...,C)$ or $DS(...,c)$, an upper case letter in the second position means a public key, and a lower case letter means a secret or private key, so that in any particular block, this refers to the fact that when a ballot is received for example, the server creates a digital signature of the ballot using the server's secret or private key. On the client 41 side, when an individual casts a ballot, the client 41 side creates a digital signature of the ballot 57 using the individual's secret or private key. Thus, both of these signatures, one created by the individual, and the other created by the server, can only have been created by the respective entities because they are both signatures that are created using the respective secret or private keys.

[0053] The individual's signature of the cast ballot $DS(B_{\text{cast}}, v)$ 57 is delivered to the server 43 side where it is combined with three other elements to form a confirmation token T 61. Specifically, the server's signature of the cast ballot $DS(B_{\text{cast}}, s)$ 53 is first combined with the individual's signature of the cast ballot $DS(B_{\text{cast}}, v)$ 57 along with a vote serial number (VSN) 58. Together the three can be seen as a confirmation sub-token ("CS"). Note that the VSN can

be like a ballot serial number and can be an arbitrary number that goes from one to infinity; it can be generated per election, and has no relationship to the voter, and is just an incidental sequence number that indicates a vote delivered in the election. The sub-token is then digitally signed by the server using the server's private key yielding $DS(CS, s)$ 59, and the four (i.e., three elements of the sub-token and the server's private signature of the subtoken) combine into an aggregation of core components which is the ballot confirmation token T 61. This allows confirmation that a particular ballot has been retained in the system and no tampering has occurred. That token is then transmitted back to the individual as confirmation token 61. The confirmation token 61 can then be encrypted with the individual's public key, thus rendering it undecipherable to anyone except the individual.

[0054] Such encryption is not mandatory but may be desirable, depending on other specific implementations of the system and method. For purposes of the disclosure herein, it is noted that while the term "token" is referred to, it could be characterized as a data element which is the signature confirmation by the server, in particular, the confirmation which has three components and a digital signature. Thus, by tying these components together, and having the server sign the components, if later on someone submits the confirmation, the server can verify that the confirmation was issued by the server, minimizing the possibility that confirmations can be forged.

[0055] On the server 43 side, the server's digital signature of the cast ballot $DS(B_{\text{cast}}, s)$ is also recorded along with the cast ballot in a database shown at block 55. If it becomes desirable to verify that all of the ballots cast in the election have been untampered with, that can be done on the server 43 side using only the contents of the database and the server ballot signatures which have been recorded in the data store.

[0056] This is further illustrated by 63 which illustrates universal verifiability. More specifically, a search is conducted through the database in a manner indexed to a vote serial number, and for each vote serial number, a ballot is reconstructed. The reconstructed ballot has no identification back to the voter and is completely anonymous. However, the ballot responses are the same as those that the individuals generated when the ballot was cast. A digital signature can then be created over the reconstructed ballot using the server's public key, and that signature can be verified against the signature that was recorded when the individual cast the ballot that

was created using the server's private key. Thus, the 63 represents the validation of the server's ballot signature created with its private key, against the server's ballot signature that was generated from the reconstructed ballot using the server's public key. This can be done for all of the ballots stored in the system, and thus, it can be verified that none of the ballots have been tampered with at any point in time after a ballot has been cast. This is referred to as universal verifiability, which refers to the property that allows for verification that for all voters none of the ballots have been forged, deleted or altered.

[0057] Consider now the case where an individual would want to connect to the system, i.e., on the server 43 side to determine if their cast ballot is properly recorded. In such a case, the individual can present vote confirmation, e.g., token 61, through a transmission 67 to the server 43 side. Of course, the individual will have to decrypt the confirmation Token 61 as previously discussed using their private key. The token 61 is then presented to the system which decomposes the token 61 into four components, which are represented by blocks 69 plus $DS_{\text{token}}(CS, s)$, and have been previously discussed with reference to blocks 59 and 61.

[0058] The digital signature on that confirmation can then be recomputed, and it can be verified that the confirmation has not been altered. Having done so, the vote serial number can then be extracted from the confirmation, and the database can be indexed to allow reconstruction of the voter's ballot exactly as cast.

[0059] Descriptors 71 and 73 illustrate what is known as individual verifiability. The server's ballot signature can be extracted out of the confirmation and validated against the server's ballot signature that is generated from the reconstructed ballot. This is demonstrated by the exchange which occurs through descriptor 71. The individual's ballot signature that was computed using the individual's private key can also be validated by the comparison represented by descriptor 73 against the digital signature that has taken over the reconstructed ballot using the voter's public key. This can be done because the communication between the individual and the server is protected, as illustrated later in Fig. 3, by a secure socket layer (SSL) connection 101 that provides temporary access to the individual's public key.

[0060] In fact, in such an implementation, the public key of the individual on the individual's end of the connection can be obtained outside of the system and once it is known what the

individual's public key is, the requirement for an LDAP certificate directory server 25 to obtain the public key can be eliminated.

[0061] As may be appreciated from the discussion of individual verifiability, in contrast to universal verifiability, individual verifiability provides two ways of verifying by using either the server's digital signature or the individual's digital signature on the ballot.

[0062] Fig. 3 shows in greater detail the general architecture shown in Fig. 1, and the data flow shown in Fig. 2. Specifically, the upper part of Fig. 3 shows the various components of the system along with boxes showing the data flow, with the lower portion of Fig. 3 showing in greater detail the various resultant tables assembled with ballots that have been cast.

[0063] A smart card reader 13 can be implemented connected to an individual's PC 11 on a smart card 103 which is read by the smart card reader 13, and can have stored thereon an individual's private (or secret) encryption key. A certificate such as an X.509 certificate for their public key, and their voter identification number which is an arbitrary sequence number generated when they register with the system, can also be stored on the smart card 103. The individual's PC 11 will typically be running a standard web browser 105, such as is commercially available from Netscape Corporation, and inside the web browser 105 is a Java interpreter 107 with the ballot 51 presented within the web browser 105. Although the ballot has been previously described, and is illustrated in Fig. 3, as an HTML or XML document, it can also take other forms such as a bit-map, if enhanced security is desired.

[0064] While a personal computer 11 has been shown with a standard commercially-available browser 105, it may also be appreciated that such browsers are often not sufficiently secure. Thus, as contemplated herein, it is also possible to implement in a manner well known to those of ordinary skill in the art, a browser which avoids the security pitfalls of currently-existing commercial browsers.

[0065] More specifically, it is common for malicious software to read/modify sensitive files on unsecure personal computers. Such unsecure personal computer platforms can include those which are Intel processor/Microsoft Windows operating system, or Apple Macintosh operating system-based. Specifically, in such commercial systems, the security ends at the client. As a result, the entire system may be vulnerable to attack by malicious software executing within the

individual's PC. With no individual platform security mechanisms, such attacks could detrimentally impact information confidentiality, integrity, authenticity, or availability within the overall system.

[0066] In order to address such a problem which may be inherent in a conventional commercial web browser, and with ballots such as an HTML or XML document ballot, a protocol can be implemented that combines anonymous context, and visual obfuscation to make it virtually impossible for malicious software to knowingly and undetectably interfere with a concurrently executing individual PC application in order to impact information confidentiality, integrity and authenticity.

[0067] In accordance with the specific implementation of such security as contemplated herein, and as will be readily apparent to those of ordinary skill in the art, cryptographic techniques, protocols and data representation can be implemented in order to increase the work required of malicious software beyond what can be accomplished within a voting session at an individual's personal computer 11. Specifically, implementation of visual obfuscation is intended to defeat the ability of malicious software to determine or alter network content. Such a technique would include but not be limited to variation of text, font, size, spacing, etc. The wording of the text associated with ballot choices, the location of graphic components, the shape of image of choice/selection targets, explicit or implicit relationships between graphical elements, and in some cases, color, can also be implemented in an obfuscating manner. In addition, a technique for converting content in HTML or XML format to image format can be employed, making it difficult, with current OCR algorithms to intercept and alter the limited ballot information being transmitted. Such OCR techniques are generally processor intensive, and thus attempts by malicious software to alter or interpret such ballot information would be readily detected.

[0068] As shown in the Fig. 3 a voter ID and the public key of the individual is transferred at 118 over an SSL connection 101 to the web server 19 on the server 43. The server side 43 includes a number of function boxes assembled as one box 109. The web server 19 which includes an execution environment for Java Servlets 111 then sends a ballot 119 obtained from a collection of ballots 113, to the individual PC 11 through a transmission 119. The individual

then marks the ballot and returns the ballot choices as shown with arrow 121. The confirmation shown as block 61 is then returned as shown by arrow 123.

[0069] It will be appreciated that within the collection 109 of function boxes, separate functionality and information is provided, for example, from the collection of election ballots 113, vote serial numbers 119, and certificates 117, characterized preferably as X.509 certificates. In the Figure, block 115 is typically the application database server 23 shown in Fig. 1 and serves to compile the election voter table, election database and optionally, a voter demographics database, all of which will be described hereafter with reference to the lower half of Fig. 3.

[0070] The lower half of Fig. 3 illustrates how the various pieces of information/data can be assembled. At block 151 it is seen that to provide voter anonymity, while retaining nonrepudiation and election integrity, cast ballots are re-signed by the server before appending to the election database. In this case, at box 151, there are reflected the choices the individual ballot items that the individual voter casts a vote for, and are stored in an election results table 157.

[0071] Table 157 conceptually includes four columns. The first column is the election in which the votes are cast. A separate database can be created per election but could also be done in this manner. The table 157 is indexed by the vote serial number (VSN), the issue or office being voted on, and the choice the individual makes. Thus, this is the table in which the ballot choices are stored. The vote serial number is also stored in a table 155 which is an election verification table. The purpose of this table is to retain the digital signatures that protect the integrity of the ballot and the election verification table has at least three columns and perhaps others. The election verification table is indexed by the vote serial number. It includes a column for time and a column for the digital signature. Optionally, demographic data could also be input into the election verification table 155 as long as this information would not, through inference or aggregation, divulge the voter's identity.

[0072] On the left side is shown the election voter table 153 which is a hashed table that in order to achieve single-vote verification, to ensure that individuals cannot vote multiple times, provides that the voter identification or ID is encrypted or hashed using a one-way transform. One option is to employ an encryption using the individual voter's public key, but any one-way transform would work. The contents of the table 153, since they are generated by a one-way hashed function are not reversible, so it is impossible to look into the table to see who voted .

However, if an individual tries to vote more than once, the hash entry will collide with any new attempt to cast a second vote, and thus, there is provided a way of detecting a subsequent voting attempt without divulging anything about the identity of the individual.

[0073] Turning now to Fig. 4, there is illustrated in block diagram form a large-scale production voting system architecture. Registration clients 201 at their personal computers are shown as able to register through connection through the global computer network 205, through a voting firewall 207, web servers 209 and application database servers 211, which may be ultimately connected to a Registrar of Voters system 221. The Registrar of Voters system 221 may provide registration information to a registration server 215 which cooperates as previously described with certificate directory servers 219, and certificate authority servers 217, and in cooperation with the other elements to allow delivery of a ballot to voting clients 203 to accomplish voting as previously described herein.

[0074] Having thus generally described the invention, the same will become better understood from the following claims in which it is set forth in a non-limiting manner.